УДК 519

О. В. Расин

ИЗОМОРФИЗМЫ ГРАФОВ С ПРОСТЫМ СПЕКТРОМ И ДРЕВЕСНЫЕ РАЗЛОЖЕНИЯ ПО РАССТОЯНИЮ

В данной работе под графом мы понимаем обыкновенный граф, т. е. неориентированный граф без петель и кратных ребер. Сразу договоримся, что для графов мы будем придерживаться терминологии и системы обозначений из книги [1]. Множество вершин графа G будем обозначать через VG, а множество ребер – через EG. Если U – подмножество множества вершин V графа G, то через $G(U)$ будем обозначать подграф графа G, порожденный множеством U.

Проблема распознавания изоморфизма графов состоит в построении полиномиального алгоритма, который для любых двух заданных графов определяет, являются ли они изоморфными. Говорят, что для данного класса графов проблема изоморфизма решена, если существует полиномиальный алгоритм, проверяющий для любой пары графов из этого класса, изоморфны ли они. В настоящее время неизвестно, существует ли полиномиальный алгоритм решения проблемы изоморфизма в классе всех графов. Но для отдельных классов такие алгоритмы найдены. Например, алгоритм Ахо–Хопкрофта–Ульмана для деревьев [2, гл. 3], алгоритм Хопкрофта–Тарьяна для планарных графов [3], алгоритм Бодлендера для графов с ограниченной древесной шириной [4], алгоритм для графов с ограниченной шириной древесного разложения по расстоянию [5], алгоритм Бабаи–Лакса для связных графов с ограниченными степенями вершин, алгоритм в классе графов, имеющих простой спектр, и алгоритм в классе графов, кратности всех собственных значений в которых ограничены [6].

Определим некоторые из классов графов, которые упоминались выше. Пусть дан обыкновенный n-вершинный граф G и пусть $A(G)$ – матрица смежности графа G. *Спектром графа* G называют набор собственных значений $\lambda_1, \lambda_2, \ldots, \lambda_n$ его матрицы смежности $A(G)$. Вообще говоря, λ_i, где $i = 1, \ldots, n$, не обязательно попарно различны. Заметим также, что λ_i, где $i = 1, \ldots, n$, называют *собственными значениями графа* G. Кратностью собственного значения λ_i, где $i = 1, \ldots, n$, называется число его вхождений в набор $\lambda_1, \lambda_2, \ldots, \lambda_n$ или, что то же самое, кратность корня λ_i в характеристическом многочлене матрицы $A(G)$.

© Расин О. В., 2010

Далее для каждого $k \in \mathbb{N}$ через $Spec_k$ будем обозначать класс графов, в которых кратность каждого собственного значения не превосходит k. Пусть k – некоторая натуральная константа. В этом случае существует полиномиальный алгоритм проверки изоморфизма графов в классе $Spec_k$ [6]. Заметим, что сложность этого алгоритма экспоненциально зависит от k.

Для класса графов $Spec_1$ Лейтоном [Leighton] был построен алгоритм проверки изоморфизма временной сложности $O(n^3)$ [6].

Пусть G – некоторый граф, а u и v – его вершины. Длина кратчайшего маршрута из вершины u в вершину v обозначается через $d_G(u, v)$ и называется *расстоянием между вершинами u и v*.

В этой работе будет рассматриваться новый класс графов, который содержит, в частности, все графы из класса $Spec_1$. Прежде чем ввести этот новый класс, нам понадобится следующее определение.

Определение. *Древесным разложением связного графа $G = (VG, EG)$ по расстоянию* называется тройка $(\{X_i : i \in I\}, T = (I, F), r)$ такая, что

1) $\bigcup\limits_{i \in I} X_i = VG$ и $X_i \cap X_j = \varnothing$ для любых $i \neq j$, где $i, j \in I$;

2) T является корневым деревом с корнем в вершине $r \in I$;

3) для каждого $v \in V$ если $v \in X_i$, то $d_G(X_r, v) = d_T(r, i)$;

4) для каждого ребра $\{v, w\} \in E$ существуют такие $i, j \in I$, что $v \in X_i$, $w \in X_j$ и либо $i = j$, либо $\{i, j\} \in F$.

Если дерево $T = (I, F)$ является простой цепью, выходящей из корня r, то такое древесное разложение графа по расстоянию будем называть *цепным разложением графа по расстоянию.*

Если множество X_r является одноэлементным, т. е. $X_r = \{u\}$, где u – некоторая вершина графа, то такое древесное разложение по расстоянию будем называть *древесным разложением по расстоянию с выделенным корнем.*

На рис. 1 приведен пример древесного разложения по расстоянию для некоторого графа.

Множества X_i, где $i \in I$, называются *компонентами древесного разложения по расстоянию.* Очевидно, что всегда существует древесное разложение графа G по расстоянию с корнем в вершине u, но оно не всегда единственно. Сразу заметим, что цепное разложение по расстоянию с заданным корнем строится однозначно, причем за линейное время. В соответствии с [5] введем следующия понятия.

Определение. Пусть $D = (\{X_i : i \in I\}, T = (I, F), r)$ – древесное разложение графа G по расстоянию. Для каждой компоненты древесного разложения

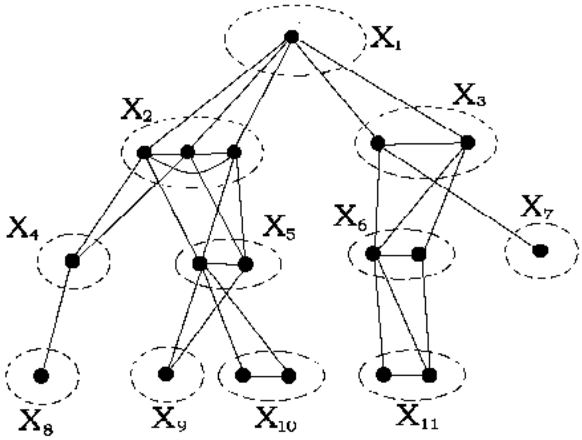

Рис. 1

X_i определим множество $V(D, X_i) = \bigcup\limits_{j \in VT_i} X_j$, где T_i – максимальное корневое поддерево дерева T с корнем в вершине i.

Определение. Пусть $D = (\{X_i : i \in I\}, T = (I, F), r)$ – древесное разложение графа G по расстоянию. D называется *минимальным*, если для каждого $i \in I$ порожденный подграф $G(V(D, X_i))$ графа G является связным.

В [5] доказано следующее утверждение.

Предложение 1. *Пусть даны граф G и множество $S \subseteq VG$. Минимальное древесное разложение графа G по расстоянию с корневым множеством S единственно, и существует алгоритм с временной сложностью $O(|EG|)$, строящий минимальное древесное разложение графа G с корневым множеством S.*

Алгоритм, который упоминается в предложении 1, можно найти в [5].

Заметим, что в [5] приведены полиномиальные алгоритмы, решающие проблему изоморфизма в следующих классах графов. Пусть k – некоторое натуральное число. Если графы обладают древесными разложениями по расстоянию, в которых каждая компонента по мощности не превосходит k (этот класс графов будем обозначать $\mathcal{TDW}_k$), то существует полиномиальный алгоритм проверки изоморфизма с временной сложностью $O((k!)^2 k^2 n^{k+2})$, где n – количество вершин в графе. В случае же если графы обладают цепными разложениями по расстоянию, в которых каждая компонента по мощности не превосходит k (этот класс графов будем обозначать $\mathcal{PDW}_k$), то существует полиномиальный алгоритм проверки изоморфизма с временной сложностью

$O((k!)^2k^2n^{k+1})$. Если же мы ограничимся графами, обладающими разложениями по расстоянию, в которых каждая компонента по мощности не превосходит k, с выделенными корнями, то получим алгоритм сложности $O((k!)^2k^2n^3)$ для древесных разложений и $O((k!)^2k^2n^2)$ для цепных разложений. Соответствующие классы будем обозначать через $\mathcal{RTDW}_k$ и $\mathcal{RPDW}_k$.

Определим класс графов $PathSpec_1$ следующим образом. Будем говорить, что граф G принадлежит классу $PathSpec_1$, если он связный и для него существует цепное разложение $P =(\{X_i : i \in I\}, T = (I, F), r)$ по расстоянию с выделенным корнем такое, что для каждого $i \in I$ граф $G(X_i)$ содержится в классе $Spec_1$. В дальнейшем такие древесные разложения по расстоянию с выделенным корнем будем называть $PathSpec_1$-разложениями. Заметим, что если граф G принадлежит классу $PathSpec_1$, то множество всех его $PathSpec_1$-разложений не обязательно одноэлементно. В этой работе будет приведен полиномиальный алгоритм для проверки изоморфизма в классе графов $PathSpec_1$.

1. Предварительные сведения о группах подстановок, изоморфизмах графов и группах автоморфизмов графов

В этой части мы приведем некоторые обозначения, определения и утверждения, которые будем использовать в дальнейшем. Неопределяемые здесь понятия для групп можно найти в книге [7].

Пусть нам даны два графа G и H. Если они изоморфны, то множество всех изоморфизмов графов G и H будем обозначать через $Iso(G, H)$. Через $Aut(G)$ обозначим группу автоморфизмов графа G. Справедливо следующее предложение [8].

Предложение 2. *Пусть графы G и H изоморфны и $g \in Iso(G, H)$ – некоторый изоморфизм графа G на граф H. Любой изоморфизм h графа G на граф H может быть представлен как суперпозиция отображений $g \circ a$, где a – элемент группы $Aut(G)$.*

Из этого предложения непосредственно вытекает

Следствие 1.1. *Если графы G и H изоморфны, то число автоморфизмов графа G равно числу изоморфизмов графа G на граф H.*

Введем следующее

Определение. Пусть G и H – непересекающиеся графы. Через $Aut(G, H)$ будем обозначать группу всех автоморфизмов графа $G \cup H$, каждый из которых либо отображает множество вершин VG графа G на множество вершин

VH графа H, а множество вершин VH графа H на множество вершин VG графа G, либо отображает множества вершин VG и VH графов G и H в себя.

Нетрудно заметить, что ограничение на множество вершин VG каждого элемента группы $Aut(G, H)$, отображающего множество вершины VG на множество вершин графа VH, является изоморфизмом графов G и H. Обратное утверждение тоже справедливо. Допуская вольность, будем говорить, что группа $Aut(G, H)$ *содержит все изоморфизмы графов* G *и* H.

Можно показать, что группа $Aut(G, H)$ определяется множеством всех изоморфизмов графа G на граф H. Для того чтобы в этом убедиться, введем

Определение. Пусть нам дана пара n-элементных множеств A и B. Рассмотрим некоторый набор биективных отображений $F = \{f_1, f_2, \ldots, f_s\}$ из множества A на множество B. Множество подстановок $\{(f_i, (f_j)^{-1}) : i, j = 1, 2, \ldots, s\}$ на $A \cup B$ будем называть *множеством подстановок на* $A \cup B$*, порождаемым множеством отображений* F, и обозначать его через $Perm(A \cup B, F)$.

Нетрудно заметить, что если графы G и H изморфны, то множество $Perm(VG \cup VH, Iso(G, H))$ является порождающим множеством группы $Aut(G, H)$.

Приведем некоторые понятия и алгоритмы теории групп подстановок, которые будут использоваться при дальнейшем изложении.

Чтобы не возникло неоднозначности толкований, договоримся, что тривиальную группу будем обозначать через I. Единицу группы G будем обозначать через id_G.

Поскольку мы используем группы не только для теоретических рассмотрений, но и для построения полиномиальных алгоритмов, естественно, что нам важно знать компактный способ задания группы.

Пусть A – группа подстановок на множестве $\{1, \ldots, n\}$ и K – некоторое ее порождающее множество. Помимо того что K должно быть не очень большого размера, мы должны уметь эффективно определять для каждой подстановки $g \in S_n$, является ли она элементом группы A.

Определение. Пусть p – некоторый полином и A – группа подстановок на множестве $\{1, \ldots, n\}$. Порождающее множество K группы A называется *порождающим множеством полиномиальной мощности*, если $|K| \le p(n)$.

Рассмотрим ряд подгрупп $I = A^{(h)} < A^{(h-1)} < \cdots < A^{(1)} < A^{(0)} = A$. Высотой ряда подгрупп будем называть количество групп в ряде, отличных от единичной группы. В данном случае высота ряда равна h. *Ширина ряда* *подгрупп* равна наибольшему среди индексов $[A^{(i-1)} : A^{(i)}]$, где $1 \leq i \leq h$. Группу A будем называть *основанием ряда подгрупп.* Пусть U_i – система представителей левых смежных классов группы $A^{(i-1)}$ по $A^{(i)}$.

Определение. Порождающее множество K группы A называется *сильным порождающим множеством для ряда подгрупп*

$$I = A^{(h)} < A^{(h-1)} < \cdots < A^{(1)} < A^{(0)} = A,$$

если для каждого $i \in \{1, 2, \ldots, h\}$ при удалении всех элементов $g \in K$, не принадлежащих A_i, получаем порождающее множество группы A_i.

Для любого ряда подгрупп всегда можно указать по крайней мере одно сильное порождающее множество, таким, например, является множество $\bigcup\limits_{i=1}^{h} U_i$. Тот факт, что это множество будет сильным порождающим множеством для ряда подгрупп $I = A^{(h)} < A^{(h-1)} < \cdots < A^{(1)} < A^{(0)} = A$, доказывается в [8].

Там же доказана следующая

Теорема 1.1. *Пусть дан ряд подгрупп*

$$I = A^{(h)} < A^{(h-1)} < \cdots < A^{(1)} < A^{(0)} = A.$$

Существует алгоритм, строящий сильное порождающее множество для данного ряда подгрупп из произвольного порождающего множества K группы A за время $O((|K| + h^2 \cdot w^2) \cdot h \cdot w \cdot (t + T))$, где h – высота ряда подгрупп; w – его ширина; t – количество операций, необходимое, чтобы вычислить произведение подстановок и найти обратную подстановку, а T равно максимальному числу операций, необходимому, чтобы для каждого $1 \leq i \leq h$ и для произвольного $g \in A^{(i-1)}$ проверить, будет ли g элементом группы $A^{(i)}$. Причем количество элементов в получаемом сильном порождающем множестве равно $\sum\limits_{i=1}^{h} [A_{i-1} : A_i]$.

Соответствующий алгоритм приводится в [8].

Если, как правило, параметр t не превосходит $O(n)$, то параметр T, вообще говоря, может экспоненциально зависеть от n. Следуя [9], введем

Определение. Подгруппа B группы A называется *полиномиально распознаваемой в A*, если существует алгоритм полиномиальной сложности, позволяющий для любого $g \in A$ установить, является ли g элементом из B.

Зафиксируем полиномы p_h и p_w. Если в ряде подгрупп

$$I = A^{(h)} < A^{(h-1)} < \dots < A^{(1)} < A^{(0)} = A$$

для каждого $i \in \{1, \dots, h\}$ группа $A^{(i)}$ полиномиально распознаваема в $A^{(i-1)}$, а высота и ширина не превосходят $p_h(n)$ и $p_w(n)$ соответственно и нам известно порождающее множество K полиномиальной мощности для группы A, то мы можем за полиномиальное время построить сильное порождающее множество группы A.

Заметим, что если нам известно сильное порождающее множество группы A, то существует полиномиальный алгоритм с временной сложностью $O((h \cdot w \cdot (t + T)))$, проверяющий для каждой подстановки $g \in S_n$, будет ли g принадлежать группе A [8].

Обратим внимание, что сильное порождающее множество строится для конкретного ряда подгрупп, т. е. для различных рядов с одним и тем же основанием могут получаться различные порождающие множества.

Определение. Пусть p_h и p_w – многочлены. Ряд подгрупп

$$I = A^{(h)} < A^{(h-1)} < \dots < A^{(r)} = B < A^{r-1} < \dots < A^1 < A^0 = A$$

называется *допустимым* для группы $B \leq S_n$, если выполняются следующие условия.

1) Нам известно для группы A порождающее множество полиномиальной мощности.

2) Высота и ширина ряда не превосходят $p_h(n)$ и $p_w(n)$ соответственно.

3) Для каждого $i \in \{1, \dots, h\}$ группа $A^{(i)}$ полиномиально распознаваема в $A^{(i-1)}$.

Нетрудно заметить, что если для группы B мы знаем некоторый допустимый ряд подгрупп, то, используя алгоритм, о существовании которого говорится в теореме 1.1, мы можем найти сильное порождающее множество полиномиальной мощности для группы B. Для этого достаточно из сильного порождающего множества группы, которая является основанием ряда, удалить элементы, не принадлежащие группе B.

В заключение приведем следующий результат из [8].

Теорема 1.2. *Пусть нам известно порождающее множество K p-группы A подстановок на n-элементном множестве X. Тогда существует алгоритм, который для любого подмножества Y множества X строит порождающее множество стабилизатора множества Y в группе A за время $O(|K| \cdot n^2 + n^6 \cdot \log_p^n)$.*

Заметим, что в результате работы алгоритма мы получаем порождающее множество мощности $O(n^2)$ для стабилизатора множества Y в группе A.

2. Изоморфизмы графов из класса $Spec_1$

Пусть G и H – графы из класса $Spec_1$. Предположим, что вершины в G и в H занумерованы числами от 1 до n. Пусть $A(G)$ и $A(H)$ – матрицы смежности графов G и H соответственно. Известно, что графы G и H изоморфны в том и только в том случае, если матрица смежности $A(G)$ графа G получается из матрицы смежности $A(H)$ графа H путем одинаковых перестановок строк и столбцов [10]. Предположим, что графы G и H изоморфны, и пусть ϕ – некоторый их изоморфизм. Поскольку нумерация вершин в графах G и H одинакова, то ϕ является биективным отображением множества $\{1, \ldots, n\}$ на себя. Иными словами, ϕ может быть задано некоторой подстановкой на множестве $\{1, \ldots, n\}$. Справедливо следующее утверждение: графы G и H изоморфны тогда и только тогда, когда существует матрица P некоторой подстановки такая, что $A(H) = P \cdot A(G) \cdot P^T$. Допуская вольность, будем называть матрицу P с таким свойством изоморфизмом графа G на граф H.

Обозначим через $\lambda_1, \lambda_2, \ldots, \lambda_n$ собственные значения матрицы смежности $A(G)$.

Пусть $u_1, \ldots, u_n$ – собственные векторы матрицы $A(G)$. Известно, что все векторы u_i попарно ортогональны и образуют базис в $\mathbb{R}^n$. Рассмотрим матрицу U, в i-м столбце которой для каждого $i = 1, 2, \ldots, n$ стоят координаты вектора u_i. Пусть D – диагональная матрица, у которой на i-м месте главной диагонали, где $i = 1, 2, \ldots, n$, стоит собственное значение λ_i, соответствующее вектору u_i. Нетрудно заметить, что справедливо матричное равенство $A(G) \cdot U = U \cdot D$.

Пусть P – $n \times n$ матрица некоторой подстановки, а u – вектор-столбец размера $n \times 1$. Через $P(u)$ будем обозначать вектор-столбец $P \cdot u$ размерности $n \times 1$. Очевидно, что вектор $P(u)$ отличается от вектора u тем, что его координаты переставлены согласно подстановке, которой отвечает P.

Очевидно, что если графы G и H изоморфны, то их спектры совпадают. Справедливо следующее

Утверждение 2.1. *Если P – изоморфизм графов G и H, то для каждого $i = 1, 2, \ldots, n$ вектор $P(u_i)$ является собственным вектором матрицы $A(H)$ с собственным значением λ_i.*

Для каждого $i = 1, 2, \ldots, n$ обозначим через w_i собственный вектор матрицы $A(H)$ с собственным значением λ_i.

По аналогии со сделанными выше построениями рассмотрим матрицу W, в i-м столбце которой для каждого $i = 1, 2, \ldots, n$ стоят координаты вектора w_i. Будем предполагать, что длины векторов w_i и u_i равны. Нетрудно заметить, что справедливо матричное равенство $A(H) \cdot W = W \cdot D$.

Определение. Будем называть квадратную матрицу E *унидиагональной*, если она является диагональной, и каждый элемент главной диагонали равен либо 1, либо -1. Множества всех унидиагональных матриц размера $n \times n$ будем обозначать через $UD(n)$

Очевидно, что множество $UD(n)$ является группой относительно умножения матриц. Причем квадрат каждого элемента из $UD(n)$ равен единичной матрице. Единичную матрицу размера $n \times n$ будем обозначать через I_n.

Следующее достаточно очевидное утверждение играет очень важную роль для построения алгоритма проверки изоморфизма в классе графов $Spec_1$.

Утверждение 2.2. *Пусть графы G и H являются графами из класса $Spec_1$ и P – матрица некоторой подстановки из S_n. Тогда P – изоморфизм графов G и H в том и только в том случае, когда существует унидиагональная матрица E размера $n \times n$ такая, что $P \cdot U = W \cdot E$.*

Очевидно, что имеется всего 2^n унидиагональных матриц размера $n \times n$. Конечно, не для всякой матрицы E из $UD(n)$ матрица $W \cdot E \cdot U^{-1}$ будет изоморфизмом графов G и H. Поскольку мы не знаем диагональные элементы матрицы E, будем считать их переменными e_i, где $i = 1, 2, \ldots, n$. Переменная e_i отвечает диагональному элементу матрицы E в позиции (i, i), где $i = 1, 2, \ldots, n$. Область определения каждой переменной e_i состоит из двух значений 1 и -1. Конечно, $e_i^2 = 1$.

Приведем схему работы алгоритма Лейтона проверки изоморфизма в классе графов $Spec_1$. Для лучшего понимания этого алгоритма мы в качестве примера будем рассматривать графы, изображенные на рис. 2.

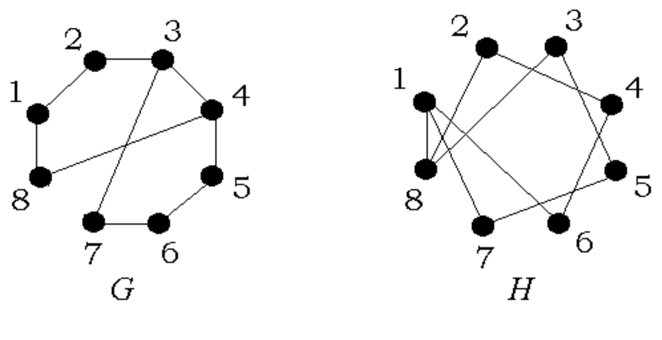

Рис. 2

Спектры графов, изображенных на рис. 2, одинаковы и состоят из следующих собственных значений (результаты вычислений мы будем приводить с точностью до третьего знака после запятой) : $\lambda_1 = 0,414$; $\lambda_2 = 1$; $\lambda_3 = 0$; $\lambda_4 = 1,414$; $\lambda_5 = -1,414$; $\lambda_6 = -1,814$; $\lambda_7 = -2$ и $\lambda_8 = 2,343$. Матрицы U и W имеют следующий вид:

$$U = \begin{pmatrix} 0,513 & 0 & 0 & -0,5 & -0,5 & -0,404 & 0 & -0,271 \\ 0,121 & -0,408 & 0,5 & -0,354 & 0,354 & 0,367 & 0,289 & -0,318 \\ -0,456 & -0,408 & 0 & 0 & 0 & -0,261 & -0,577 & -0,473 \\ -0,456 & 0,408 & 0 & 0 & 0 & -0,261 & 0,577 & -0,473 \\ 0,121 & 0,408 & 0,5 & 0,354 & -0,354 & 0,367 & -0,289 & -0,318 \\ 0,513 & 0 & 0 & 0,5 & 0,5 & -0,404 & 0 & -0,271 \\ 0,121 & -0,408 & -0,5 & 0,354 & -0,354 & 0,367 & 0,289 & -0,318 \\ 0,121 & 0,408 & -0,5 & -0,354 & 0,354 & 0,367 & -0,289 & -0,318 \end{pmatrix}$$

и

$$W = \begin{pmatrix} 0,456 & 0,408 & 0 & 0 & 0 & -0,261 & -0,577 & 0,473 \\ -0,121 & -0,408 & 0,5 & 0,354 & -0,354 & 0,367 & -0,289 & 0,318 \\ -0,121 & -0,408 & -0,5 & -0,354 & 0,354 & 0,367 & -0,289 & 0,318 \\ -0,513 & 0 & 0 & 0,5 & 0,5 & -0,404 & 0 & 0,271 \\ -0,513 & 0 & 0 & -0,5 & -0,5 & -0,404 & 0 & 0,271 \\ -0,121 & 0,408 & -0,5 & 0,354 & -0,354 & 0,367 & 0,289 & 0,318 \\ -0,121 & 0,408 & 0,5 & -0,354 & 0,354 & 0,367 & 0,289 & 0,318 \\ 0,456 & -0,408 & 0 & 0 & 0 & -0,261 & 0,577 & 0,473 \end{pmatrix}.$$

Очевидно, если матрица P для некоторой унидиагональной матрицы E удовлетворяет соотношению $P \cdot U = W \cdot E$, то можно заключить следующее: если вершина i графа G переходит в вершину j графа H, то для каждого $i_1 = 1, 2, \ldots, n$ модуль i-й координаты вектора u_{i_1} совпадает с модулем j-й координаты вектора w_{i_1} (это вытекает из утверждения 2.1 и того, что $|u_{i_1}| = |w_{i_1}|$). Используя это, мы можем следующим образом разбить множество вершин графа G на непересекающиеся классы $C_1, C_2, \ldots, C_k$ и множество вершин графа H на непересекающиеся классы $C'_1, C'_2, \ldots, C'_{k'}$. Вершины i и j графа G попадают в одинаковый класс, если для каждого $i_1 = 1, 2, \ldots, n$ модуль i-й координаты вектора u_{i_1} совпадает с модулем j-й координаты вектора u_{i_1}. Аналогичное требование выполняется для вершин графа H и классов $C'_1, C'_2, \ldots, C'_{k'}$. Ясно, что если $k \neq k'$, то графы G и H не изоморфны. Поэтому предположим, что $k = k'$. Предположим также, что классы занумерованы таким образом, что для каждого $j_1 \in \{1, \ldots, k\}$, если вершина i графа G лежит в C_{j_1} и вершина j графа H лежит в C'_{j_1}, то для $i_1 = 1, 2, \ldots, n$ модуль i-й координаты вектора u_{i_1} совпадает с модулем j-й координаты вектора w_{i_1}. Предположим, что $|C_{j_1}| = |C'_{j_1}|$ для каждого $j_1 \in \{1, \ldots, k\}$, ибо в противном случае графы G и H, очевидно, не изоморфны.

Для графов, изображенных на рис. 2, мы получаем следующее разбиение на классы: $C_1 = (1, 6)$, $C_2 = (2, 5, 7, 8)$, $C_3 = (3, 4)$ для графа G и $C'_1 = (4, 5)$, $C'_2 = (2, 3, 6, 7)$, $C'_3 = (1, 8)$ для графа H.

Дальнейшие действия направлены на то, чтобы сократить перебор вариантов в каждом из полученных классов, т. е. разбить полученные классы на более мелкие подклассы.

Для некоторого $j_1 \in \{1, \ldots, k\}$ рассмотрим классы C_{j_1} и C'_{j_1}. Далее будем обозначать их через C и C' соответственно. Для каждого вектора u_{i_1}, где $i_1 = 1, 2, \ldots, n$, обозначим через $C[u_{i_1}]$ координаты вектора u_{i_1} с номерами из C. Аналогичное обозначение $C'[w_{i_1}]$ введем для каждого вектора w_{i_1}, где $i_1 = 1, 2, \ldots, n$. Очевидно, что числа в $C[u_{i_1}]$ и $C'[u_{i_1}]$ могут различаться соответственно только знаками, а их абсолютные величины одинаковы. Далее будем мыслить $C[u_{i_1}]$ и $C'[w_{i_1}]$ для каждого $i_1 = 1, 2, \ldots, n$ как вектор соответствующих знаков.

Например, для графов на рис. 2 в этих обозначениях $C_2[u_1] = (+, +, +, +)$, $C'_1[w_5] = (+, -)$, а $C_3[u_4] = (0, 0)$.

Конечно, координаты вектора над классом могут оказаться нулевыми. Но этот случай, как будет видно из дальнейшего, не может дать нам никакой новой информации, и поэтому мы не будем его рассматривать. Заметим только, что для каждого вектора существует класс, над которым он не является нулевым. Это следует из того, что собственные векторы по определению ненулевые.

Рассмотрим u_{i_1} и w_{i_1} для некоторого $i_1 = \{1, 2, \ldots, n\}$. Рассмотрим векторы знаков $C[u_{i_1}]$ и $C'[w_{i_1}]$. Обозначим через k_+ и k_- – количество плюсов и минусов в $C[u_{i_1}]$. Аналогично, обозначим через k'_+ и k'_- количество плюсов и минусов в $C'[u_{i_1}]$. Возможны следующие случаи.

1) $k_- = 0$, $k_+ \neq 0$. В этом случае (равно как и в случае $k_+ = 0$, $k_- \neq 0$) вектор знаков $C[u_{i_1}]$ будем называть *знакопостоянным*. Из соотношения $PW = VE$ легко увидеть, что графы G и H могут быть изоморфными, только если либо $k'_- = 0$, либо $k'_+ = 0$. Например, в случае $k'_+ = 0$ получаем, что $e_{i_1} = -1$, а в случае $k'_- = 0$ получаем $e_{i_1} = 1$.

2) $k_+ = 0$, $k_- \neq 0$. Этот случай аналогичен предыдущему.

3) $k_+ > k_-$ и $k_- \neq 0$. При этих условиях G и H могут быть изоморфны только в случае, когда либо $k_+ = k'_+$ (автоматически $k_- = k'_-$), либо $k_+ = k'_-$ (автоматически $k_- = k'_+$). Например, предположим, что мы имеем случай $k_+ = k'_+$. Тогда мы получаем $e_{i_1} = 1$. Кроме того, мы можем разбить классы C и C' на подклассы C_+ и C_-, C'_+ и C'_-. В $C_+(C_-)$ попадают вершины из C, имеющие соответственно положительные (отрицательные) координаты. Аналогично в $C'_+(C'_-)$ попадают вершины из C', имеющие соответственно положительные (отрицательные) координаты. Очевидно, что вершины из

C_+ должны переходить в вершины из C'_+, а вершины из C_- должны переходить в вершины из C'_-. Случай $k_+ = k'_-$ разбирается аналогично. Только мы получаем, что $e_{i_1} = -1$ и что вершины из C_+ переходят в вершины из C'_-, а вершины из C_- переходят в вершины из C'_+.

4) $k_- > k_+$ и $k_+ \neq 0$. Аналогичен предыдущему случаю.

5) $k_+ = k_- \neq 0$. Если $k'_+ = k'_- = 0$ или $k'_+ \neq k'_-$, то, очевидно, графы G и H не изоморфны. Если же $k'_+ = k'_- \neq 0$, то, в отличие от предыдущих случаев, мы ничего не можем сказать о знаке e_{i_1} и не можем разбить классы C и C' на подклассы. В таких случаях вектор знаков $C[u_{i_1}]$ будем называть *сбалансированным.*

6) $k_+ = k_- = 0$. Очевидно, что при этих условиях мы не можем разбить классы C и C' на подклассы и определить знак e_{i_1}.

Для графов, изображенных на рис. 2, получаем следующее. Рассматриваем классы C_1 и C'_1. Для векторов u_1 и w_1 имеем случай 1 и получаем, что $e_1 = -1$. Аналогично (случай 2) получаем, что $e_6 = 1$ и $e_8 = -1$. Для остальных пар векторов имеют место случаи 5 и 6. Далее рассматриваем классы C_2 и C'_2. Найденные ранее значения e_1, e_6 и e_8 удовлетворяют соответствующим парам векторов знаков над C_2 и C'_2. В противном случае мы бы получили, что графы, изображенные на рис. 2, не изоморфны, так как не существовало бы никакой унидиагональной матрицы E, для которой $P \cdot U = W \cdot E$. Для остальных пар векторов имеет место случай 5. Рассмотрение классов C_3 и C'_3 также не дает никаких противоречий, но и не позволяет получить ничего нового.

Определим операцию произведения $\bigotimes$ на векторах знаков одинакового размера как покомпонентное умножение. Иными словами, при произведении двух ненулевых векторов знаков получаем вектор знаков такого же размера, и его компонента $+$, если соответствующие компоненты исходных векторов имеют одинаковый знак, в противном случае получаем компоненту $-$. Если хотя бы один из перемножаемых векторов нулевой, то в результате получаем нулевой вектор. Например,

$$C_2[u_2] \bigotimes C_2[u_3] = (-,+,-,+) \bigotimes (+,+,-,-) = (-,+,+,-).$$

Пусть I – некоторое подмножество множества $\{1, 2, \ldots, n\}$. Предполагаем, что для любого $i_1 \in I$ вектора $C[u_{i_1}]$ и $C'[w_{i_1}]$ ненулевые. Будем обозначать произведение $\bigotimes\limits_{i_1 \in I} C[u_{i_1}]$ через $C[U^I]$. Через e^I обозначим произведение $\prod\limits_{i_1 \in I} e_{i_1}$. Если мы рассмотрим векторы знаков $C[U^I]$ и $C'[W^I]$, то из соотношения $PU = WE$ нетрудно понять, что мы получим случаи, аналогичные случаям 1–6, рассмотренным выше. Единственное отличие будет состоять в том, что в случае несбалансированного вектора знаков мы получим не значение переменной e_{i_1}, а некоторое выражение $\prod\limits_{i_1 \in I} e_{i_1} = 1$ или $\prod\limits_{i_1 \in I} e_{i_1} = -1$.

Назовем класс C *несводимым*, если для любого подмножества I множества $\{1, 2, \ldots, n\}$ такого, что для любого $i_1 \in I$ вектор $C[u_{i_1}]$ ненулевой; вектор $C[u^I]$ является сбалансированным или знакопостоянным. Справедливо следующее утверждение.

Предложение 3. *Пусть C – несводимый класс. Обозначим через I множество всех индексов i_1 из $\{1, 2, \ldots, n\}$ таких, что $C[u_{i_1}]$ сбалансированный. Тогда существует подмножество $S \subseteq I$ такое, что для любого $i_1 \in I$ найдется подмножество $S' \subseteq S$, $C[u_{i_1}] = \alpha \bigotimes\limits_{i_2 \in S'} C[u_{i_2}]$, где $\alpha \in \{-, +\}$. Причем* $|S| \leq \log_2 |C|$.

Доказательство этого результата Лейтона мы приводить не будем. Заметим только, что оно конструктивное, т. е. известен алгоритм, который в случае, если C – несводимый класс, находит такое множество S, либо мы получаем, что класс C не является несводимым, и тогда разбиваем класс C аналогично тому, как это демонстрировалось выше, и рассматриваем подклассы класса C. Заметим также, что сложность этого алгоритма не превосходит $O(|C|^2)$.

Следствие 2.1. *Пусть C – несводимый класс. Обозначим через I множество всех индексов i_1 из $\{1, 2, \ldots, n\}$ таких, что $C[u_{i_1}]$ сбалансированный. Тогда существует подмножество $S \subseteq I$ такое, что для любого $i_1 \in I$ найдется подмножество $S' \subseteq S$, $e_{i_1} = e^{S'}$. Причем $|S| \leq \log_2 |C|$.*

Доказательство. Пусть C' – класс, который соответствует классу C в графе H. Пусть $i_1 \in I$. Тогда имеем $e_{i_1} \cdot C'[w_{i_1}] = C[P(u_{i_1})]$. Из предложения 3 вытекает, что $C[P(u_{i_1})] = C[P(\alpha U^{S'})]$, где $\alpha \in \{-, +\}$. В наших обозначениях выполняется $P(U^{S'}) = e^{S'} W^{S'}$. Таким образом, $e_{i_1} \cdot C'[w_{i_1}] = \alpha e^{S'} W^{S'}$. Поскольку $C'[w_{i_1}] = \alpha W^{S'}$, заключаем, что $e_{i_1} = e^{S'}$.

В результате мы получаем систему уравнений относительно неизвестных e_i, где $i = 1, 2, \ldots, n$, которая является линейной системой относительно операции умножения над полем $\mathbb{Z}_2$.

В результате работы алгоритма для каждого класса C_i, где $i = 1, 2, \ldots, k$, мы получаем систему из $O(n)$ линейных уравнений относительно неизвестных e_i, где $i = 1, 2, \ldots, n$. Из следствия 2.1 вытекает, что число решений системы линейных уравнений для каждого класса C_i, где $i = 1, 2, \ldots, k$, не превосходит $O(n)$. Действительно, мощность базиса пространства решений не превосходит

$\log_2 |C|$, а поскольку область определения каждой переменной состоит из двух значений, то всего решений не больше $2^{\log_2 |C|} = |C| \le n$.

Поскольку классов C_i не более чем n, в каждом из них $O(n)$ уравнений и количество решений над каждым классом не превосходит $O(n)$, мы получаем линейную систему из $O(n^2)$ уравнений с n неизвестными. Применяем метод исключения Гаусса. Если система несовместна, то графы G и H неизоморфны. В противном случае существует решение $\{e_1, e_2, \dots, e_n\}$, с помощью которого, как следует из утверждения 2.2, мы можем получить изоморфизм $P = W \cdot E \cdot U^{-1}$. Причем каждому изоморфизму графов G и H соответствует некоторое решение полученной системы. Справедливо и обратное

Утверждение 2.3. *Существует алгоритм, который для каждой пары графов из класса $Spec_1$ устанавливает, изоморфны они или нет за время $O(n^3)$ в худшем случае.*

Для графов, изображенных на рис. 2, получаем следующее. Рассматриваем классы C_1 и C_1'. Построим для них множество S_1 (аналог множества S из предложения 2.1 для этих классов). В обозначениях предложения 2.1 имеем, что $I_1 = \{4, 5\}$. Добавляем в S_1 элемент 4. Поскольку очевидно, что $C_1[u_4] = C_1[u_5]$, получаем уравнение $e_5 = e_4$. В данном случае система состоит из одного уравнения. Рассматриваем классы C_1 и C_1'. Имеем, что $I_1 = \{2, 3, 4, 5, 7\}$. Добавляем к S_2 элемент 2. Очевидно, $C[u_3]$ не выражается через $C[u_2]$, поэтому в S_2 добавляем 3. Вектор $C[u_4]$ не выражается ни через $C[u_2]$, ни через $C[u_3]$, но легко проверить, что $C[u_4] = -C[u_2] \bigotimes C[u_3]$. Отсюда следует, что $e_4 = e_2 \cdot e_3$. Аналогично получаем $C[u_5] = C[u_2] \bigotimes C[u_3]$ и $e_5 = e_2 \cdot e_3$. Нетрудно заметить, что $C[u_7] = -C[u_2]$ и $e_7 = e_2$. Рассматривая классы C_3 и C_3', находим $e_7 = e_2$.

В нашем случае мы получаем три системы линейных уравнений. Отбрасывая одинаковые уравнения, в итоге имеем систему.

$$\begin{cases} e_4 = e_2 e_3, \\ e_5 = e_2 e_3, \\ e_7 = e_2. \end{cases}$$

Напомним, что e_i, которые не вошли в систему, были найдены раньше. Эта система имеет решение. Последнее означает, что графы изоморфны. Очевидно, число решений равно четырем. Перечислим все четыре изоморфизма (мы их представим в виде подстановок):

$$f_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 2 & 8 & 1 & 7 & 4 & 3 & 6 \end{pmatrix}, \qquad f_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 3 & 8 & 1 & 6 & 4 & 2 & 7 \end{pmatrix},$$
$$f_3 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 7 & 1 & 8 & 2 & 5 & 6 & 3 \end{pmatrix}, \qquad f_4 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 6 & 1 & 8 & 3 & 5 & 7 & 2 \end{pmatrix}.$$

Вообще говоря, количество решений получаемой системы линейных уравнений может экспоненциально зависеть от n. Поэтому (это будет понятно при описании алгоритма проверки изоморфизма графов в классе $PathSpec_1$) нам понадобятся некоторые структуры, которые могут быть заданы количеством элементов, полиномиально зависящим от n. Эти структуры будут содержать информацию обо всех изоморфизмах графов класса $Spec_1$. В качестве таких структур мы будем использовать группы подстановок.

Для каждого класса C_i, где $i \in \{1, 2, \dots, k\}$, мы имеем систему из $O(n)$ уравнений относительно неизвестных e_i, $i \in \{1, 2, \dots, n\}$, причем эта система имеет $O(n)$ решений. Используя этот факт, мы можем получить все возможные отображения множества вершин класса C_i графа G на множество вершин класса C_i' графа H. Обозначим множество таких отображений через $P[C_i]$ для каждого $i = 1, 2, \dots, k$.

Например, для графов на рис. 2 множество $P[C_1]$ состоит из двух отображений: одно переводит 1 в 5 и 6 в 4, а другое – 1 в 4 и 6 в 5.

Очевидно, что любой изоморфизм графов G и H является элементом прямого произведения $P[C_1] \times P[C_2] \times \dots \times P[C_k]$. Обратное утверждение, вообще говоря, не всегда справедливо. Заметим также, что число отображений в $P[C_1] \times P[C_2] \times \dots \times P[C_k]$ может экспоненциально зависеть от n.

Все приведенные выше факты нам понадобятся для решения следующей задачи.

Проблема 1. *Пусть G и H – изоморфные графы из класса $Spec_1$. Найти порождающее множество K полиномиальной мощности для группы автоморфизмов $Aut(G, H)$.*

Рассмотрим множество $K' = Perm(VG \cup VH, P[C_1] \times P[C_2] \times \dots \times P[C_k])$ подстановок на множестве вершин графа $G \cup H$. Пусть A' – группа подстановок на множестве вершин графа $G \cup H$, порожденная множеством K'. Из предложения 2 вытекает, что все элементы группы $Aut(G, H)$ содержатся в группе A'. Конечно, мощность множества K' может экспоненциально зависеть от n. Для дальнейших же рассмотрений нам понадобится порождающее множество полиномиальной мощности группы A'. Рассмотрим множество $K_i' = Perm(C_i \cup C_i', P[C_i] \bigotimes P[C_i'])$. Пусть A_i' – группа, порождаемая

K'_i. Обозначим множество подстановок $\{(id_{A'_1}, id_{A'_2}, \dots, g_i, \dots, id_{A'_k}) : g_i \in K'_i\}$ через K_i. Очевидно, что мощность множества K_i есть $O(n)$. Легко заметить, что множество $K = \bigcup\limits_{i=1}^{k} K_i$ является порождающим множеством группы A', количество элементов в нем $O(n^2)$.

Пусть $E_{i,j} = \{\{v_1, v_2\} \in EG \cup EH : v_1 \in {}_i \cup {}'_i, v_2 \in {}_j \cup {}'_j\}$ – подмножество множества ребер графа $G \cup H$, соединяющих вершины из $C_i \cup C'_i$ с вершинами из $C_j \cup C'_j$ в графе $G \cup H$. Положим $F_1 = \varnothing$ и $r = \binom{k}{2} + 1$.

Определим последовательности $G_k = (V, F_k)$, где $1 \leq i \leq r$, следующим образом:

$$
\begin{aligned}
&G_1 = (VG \cup VH, F_1),\\
&G_2 = (VG \cup VH, F_2) = (VG \cup VH, F_1 \cup E_{1,1}),\\
&G_3 = (VG \cup VH, F_3) = (VG \cup VH, F_2 \cup E_{1,2}),\\
&\qquad\qquad\qquad\qquad \dots\dots\dots\dots\dots\dots\\
&G_{s+1} = (VG \cup VH, F_{s+1}) = (VG \cup VH, F_s \cup E_{1,s}),\\
&G_{s+2} = (VG \cup VH, F_{s+2}) = (VG \cup VH, F_{s+1} \cup E_{2,2}),\\
&G_{s+3} = (VG \cup VH, F_{s+3}) = (VG \cup VH, F_{s+2} \cup E_{2,2}),\\
&\qquad\qquad\qquad\qquad \dots\dots\dots\dots\dots\dots\\
&G_r = (VG \cup VH, F_r) = (VG \cup VH, F_{r-1} \cup E_{s,s}) = G \cup H.
\end{aligned}
$$

Для каждого $1 \leq i \leq r$ группу автоморфизмов графа G_i, каждый элемент которой либо отображает множества VG и VH друг на друга, либо стабилизирует множества VG и VH, будем обозначать через $A^{(i)}$. Очевидно, что $A_1 = A'$, а $A_r = Aut(G, H)$. Нетрудно заметить, что группы $A^{(i)}$, где $i \in \{1, 2, \dots, r\}$, образуют ряд $A^{(1)} \geq A^{(2)} \geq \dots \geq A^{(r)}$. Для каждого $1 \leq j \leq k$ обозначим через $B^{(j)}$ поточечный стабилизатор множества C_j в группе $Aut(G, H)$ и положим $B^{(0)} = Aut(G, H))$.

Таким образом, для группы $Aut(G, H)$ мы имеем ряд

$$A^{(1)} \geq A^{(2)} \geq \dots \geq A^{(r)} = Aut(G, H) = B^{(0)} \geq B^{(1)} \geq \dots \geq B^{(k)}. \tag{1}$$

Чтобы убедиться в существовании полиномиального алгоритма, решающего задачу 1, достаточно преверить, что ряд 1 является допустимым для $Aut(G, H)$.

Очевидно, что высота ряда 1 не превосходит $n^2 + n$. Оценим ширину этого ряда.

Предложение 4. *Для каждого* $1 < i \leq r$ *индекс группы* $A^{(i-1)}$ *по подгруппе* $A^{(i)}$ *не превосходит* $O(n^2)$.

Доказательство. Рассмотрим графы G_{i-1} и G_i из построенной выше последовательности графов, которые соответствуют группам $A^{(i-1)}$ и $A^{(i)}$. Очевидно, что граф G_i отличается от графа G_{i-1} только тем, что к нему добавлено множество ребер $E_{i',j'}$ для некоторых $i_1, j_1 \in \{1, 2, \ldots, k\}$. Очевидно, что если элементы g и h группы $A^{(i-1)}$ поточечно стабилизируют множества $C_{i_1} \cup C'_{i_1}$ и $C_{j_1} \cup C'_{j_1}$, то они лежат в одном и том же смежном классе $A^{(i-1)}$ по подгруппе $A^{(i)}$. Следовательно, число смежных классов не превосходит числа всех возможных действий группы $A^{(i-1)}$ на множестве $(C_{i_1} \cup C'_{i_1}) \times (C_{j_1} \cup C'_{j_1})$. Нетрудно заметить, что это число не превосходит

$$2 \cdot |Perm(C_{i_1} \cup C'_{i_1}, P[C_{i_1}]| \times P[C'_{i_1}])| \cdot |Perm(C_{j_1} \cup C'_{j_1}, P[C_{j_1}]\ P[C'_{j_1}])| \leq$$
$$\leq 2 \cdot (2n)^2 = O(n^2).$$

Аналогичным образом доказывается следующее

Предложение 5. *Для каждого* $0 < i \leq k$ *индекс группы* $B^{(i-1)}$ *по подгруппе* $B^{(i)}$ *не превосходит* $O(n)$.

Предложение 6. *Каждая группа в ряду* 1 *полиномиально распознаваема в предыдущей группе.*

Доказательство. Рассмотрим графы G_{i-1} и G_i из построенной выше последовательности графов, которые соответствуют группам $A^{(i-1)}$ и $A^{(i)}$. Очевидно, что граф G_i отличается от графа G_{i-1} только тем, что к нему добавлено множество ребер $E_{i',j'}$ для некоторых $i', j' \in \{1, 2, \ldots, k\}$. Таким образом, для $g \in A^{(i-1)}$ нам нужно проверить, сохраняет ли g ребра между вершинами из $C_{i'}$ и $C_{j'}$. Очевидно, эту проверку можно осуществить за время $O(|C_{i'}| \cdot |C_{j'}|) = O(n^2)$.

Для групп B_j, где $0 < j \leq k$, утверждение очевидно. Здесь проверку можно осуществить за время $O(n)$.

Поскольку нами уже построено порождающее множество мощности $O(n^2)$ для группы, которая является основанием ряда 1, справедливо следующее

Предложение 7. *Ряд* 1 *является допустимым для подгруппы* $Aut(G, H)$.

Используя теорему 1.1 и применяя алгоритм, который упоминается в ней, к ряду 1, получаем следующую теорему.

Теорема 2.1. *Пусть* G *и* H *– изоморфные графы из класса* $Spec_1$. *Существует алгоритм, строящий сильное порождающее множество полиномиальной мощности группы* $Aut(G, H)$ *за время* $O(n^{14})$. *Мощность этого порождающего множества не превосходит* $O(n^4)$.

Алгоритм, о существовании которого говорится в формулировке теоремы, мы будем обозначать $FindAutGraph(G, H)$. Будем предполагать, что при такой записи G и H суть графы из класса $Spec_1$, $FindAutGraph$ является функцией и возвращает сильное порождающее множество группы $Aut(G, H)$.

В заключение докажем следующую теорему.

Теорема 2.2. *Пусть G и H – изоморфные графы из класса $Spec_1$. Группа $Aut(G, H)$ является 2-группой.*

Доказательство. В [11] замечено, что группа автоморфизмов графа из класса $Spec_1$ является 2-группой. Это можно показать и непосредственно. Докажем, например, что группа автоморфизмов графа G является 2-группой. Из утверждения 2.2 следует, что если P – автоморфизм графа G, то выполняется матричное равенство $P \cdot U = W \cdot E$, где E – унидиагональная матрица. Очевидно, что $E^2 = I$. Поскольку все собственные векторы матрицы $A(G)$ линейно независимы, то матрица U невырождена. Поэтому $P = W \cdot E \cdot U^{-1}$, откуда следует, что $P^2 = I$. Следовательно, порядок каждого элемента группы $Aut(G)$ равен 2, т. е. $Aut(G)$ является 2-группой. Положим $|Aut(G)| = 2^s$. Тогда из следствия 1.1 получаем, что $|Iso(G, H))| = Aut(H) = 2^s$. Поскольку $|Aut(G, H)| = |Aut(G)| \cdot |Aut(H)| + |Iso(G, H)| \cdot |Iso(H, G)| = 2^{2s} + 2^{2s} = 2^{2s+1}$. Теорема доказана.

3. Изоморфизмы графов в классе $PathSpec_1$

Определение. Пусть u – вершина графа G, а v – вершина графа H. Графы G и H будем называть $[u, v]$-*изоморфными*, если существует изоморфизм графа G на граф H, отображающий вершину u на v

Пусть G – граф из класса $PathSpec_1$. Нетрудно заметить, что у графа G может существовать несколько вершин u, которые могут быть корнями $PathSpec_1$-разложений. $PathSpec_1$-разложение $P =(\{X_i : i \in I\}, T = (I, F), u)$ графа G с корнем в вершине u будем обозначать через P_u^G.

Пусть H – граф из класса $PathSpec_1$. Будем говорить, что $PathSpec_1$-разложения P_u^G и P_v^H графов G и H с корнями в вершинах u и v соответственно *изоморфны*, если $[u, v]$-изоморфны графы G и H.

Пусть G и H – графы из класса $PathSpec_1$. Зафиксируем некоторую вершину u графа G такую, что существует $PathSpec_1$-разложение P_u^G графа G. Ясно, что графы G и H будут изоморфны тогда и только тогда, когда найдется такая вершина v графа H, что графы G и H будут $[u, v]$-изоморфны (очевидно, что в этом случае существует $PathSpec_1$-разложение P_v^H графа H).

Таким образом, для того чтобы установить за полиномиальное время, являются ли G и H изоморфными графами, нам достаточно найти полиномиальный алгоритм, решающий следующую задачу.

Проблема 2. *Даны два произвольных графа G и H из класса $PathSpec_1$, u – вершина графа G, для которой существует $PathSpec_1$-разложение P_u^G графа G, а v – вершина графа H, для которой существует $PathSpec_1$-разложение P_v^H графа H. Установить, существует ли $[u, v]$-изоморфизм графа G на граф H.*

Пусть $P^G = P_u^G = (X_1, \ldots, X_t)$ и $P^H = P_v^H = (Y_1, \ldots, Y_{t'})$, причем $X_1 = \{u\}$ и $Y_1 = \{v\}$, и для каждого $i = 1, \ldots, t$ в X_i содержатся все вершины из G, находящиеся на расстоянии $i-1$ от корня, а для каждого $j = 1, \ldots, t'$ в Y_j содержатся все вершины из H, находящиеся на расстоянии $j-1$ от корня. Если $t \neq t'$, то, очевидно, P^G и P^H не изоморфны. Поэтому будем считать, что $t = t'$.

Для каждого i такого, что $i = 1, \ldots, t-1$, обозначим через G_i подграф $G(X_i \cup X_{i+1})$ графа G. Аналогичное обозначение H_i введем для подграфов $H(Y_i \cup Y_{i+1})$ графа H.

Очевидно, что если существует отображение $f : VG \to VH$, ограничение которого на $X_i \cup X_{i+1}$ для каждого $i = 1, \ldots, t-1$ является изоморфизмом графов G_i и H_i таким, что $f(X_i) = Y_i$ и $f(X_{i+1}) = Y_{i+1}$, то P^G и P^H изоморфны. Из определения изоморфизма $PathSpec_1$-разложений следует, что справедливо и обратное.

Опишем основные идеи алгоритма. Сначала рассматриваем графы $G(X_t)$ и $H(Y_t)$. Проверяем, изоморфны они или нет. Если графы $G(X_t)$ и $H(Y_t)$ не изоморфны, то графы G и H не будут $[u, v]$-изоморфными и алгоритм заканчивает работу. Если графы $G(X_t)$ и $H(Y_t)$ изоморфны, то с помощью функции $FindAutGraph$ (эта функция упоминается после теоремы 2.1) находим сильное порождающее множество $\overline{K_t}$ группы $Aut(G(X_t), H(Y_t))$. Группу $Aut(G(X_t), H(Y_t))$ обозначим через $\overline{A_t}$.

Дальнейшие действия можно разбить на $t-1$ итерацию.

Перед началом выполнения h_1-й итерации $1 \leq h_1 \leq t-1$ нам известно порождающее множество $\overline{K_{t-h_1+1}}$ группы $\overline{A_{t-h_1+1}}$. Для $h_1 = 1$ мы уже нашли $\overline{K_t}$ и $\overline{A_t}$. Строение групп $\overline{A_{t-h_1+1}}$ для $h_1 > 1$ будет ясно из дальнейших построений. Здесь заметим только, что группа $\overline{A_{t-h_1+1}}$ содержит все изоморфизмы графов $G(X_{t-h_1+1})$ и $H(Y_{t-h_1+1})$, продолжаемые до изоморфизмов графов $G\left(\bigcup\limits_{i=t-h_1+1}^{t} X_i\right)$ и $H\left(\bigcup\limits_{i=t-h_1+1}^{t} Y_i\right)$. Кроме того, группа $\overline{A_{t-h_1+1}}$ является 2-группой.

На h_1-й итерации рассматриваем граф $G_{t-h_1} \cup H_{t-h_1}$. Проверяем, изоморфны или нет графы $G(X_{t-h_1})$ и $H(Y_{t-h_1})$. Если они не изоморфны, то графы G и H не будут $[u, v]$-изоморфными и алгоритм заканчивает работу. Если графы $G(X_{t-h_1})$ и $H(Y_{t-h_1})$ изоморфны, то, применяя функцию $FindAutGraph$, находим сильное порождающее множество K'_{t-h_1} группы $Aut(G(X_{t-h_1}), H(Y_{t-h_1}))$. Обозначим эту группу через A'_{t-1}.

Рассмотрим группу $A = A'_{t-h_1} \times \overline{A_{t-h_1+1}}$. Порождающее множество K этой группы мы можем получить из порождающих множеств K'_{t-h_1} и $\overline{K_{t-h_1+1}}$. Положим

$$K = \left\{ \left(g, id_{\overline{A_{t-h_1+1}}}\right) : g \in K'_{t-h_1} \right\} \bigcup \left\{ \left(id_{A'_{t-h_1}}, g\right) : g \in \overline{K_{t-h_1+1}} \right\},$$

где $id_{\overline{A_{t-h_1+1}}}$ – единица группы $\overline{A_{t-h_1+1}}$, а $id_{A'_{t-h_1}}$ – единица группы A'_{t-h_1}. Очевидно, из теоремы 2.2 следует, что A является 2-группой. Группу A можно рассматривать и как группу подстановок на множестве вершин графа $G_{t-h_1} \cup H_{t-h_1}$, и как группу подстановок на множестве упорядоченных пар (v_1, v_2), где v_1 – вершина из $X_{t-h_1} \cup Y_{t-h_1}$, а v_2 – вершина из $X_{t-h_1+1} \cup Y_{t-h_1+1}$. Обозначим через $E^G_{t-h_1}$ множество всех ребер графа G_{t-h_1}, которые соединяют вершины из X_{t-h_1} с вершинами из X_{t-h_1+1}; аналогично через $E^H_{t-h_1}$ множество всех ребер графа H_{t-h_1}, которые соединяют вершины из Y_{t-h_1} с вершинами из Y_{t-h_1+1}. Будем рассматривать множество $E^G_{t-h_1}$ как множество упорядоченных пар (u_1, u_2), где $u_1 \in X_{t-h_1}$, а $u_2 \in X_{t-h_1+1}$. Аналогичное допущение сделаем для множества ребер $E^H_{t-h_1}$ графа H_{t-h_1}. Будем предполагать, что $|E^G_{t-h_1}| = |E^H_{t-h_1}|$, так как в противном случае графы G и H не будут $[u, v]$-изоморфными. С помощью алгоритма, о существовании которого говорится в теореме 1.2, мы можем найти порождающее множество K_E стабилизатора множества ребер $E^G_{t-h_1} \cup E^H_{t-h_1}$ графа $G_{t-h_1} \cup H_{t-h_1}$ в группе A. Обозначим этот стабилизатор через A_E. Очевидно, поскольку любой его элемент g является элементом группы A, g может рассматриваться и как подстановка на множестве вершин графа $G_{t-h_1} \cup H_{t-h_1}$.

Нетрудно заметить, что если элемент g группы A_E отображает множество вершин X_{t-h_1} на Y_{t-h_1}, то он отображает Y_{t-h_1} на X_{t-h_1}, X_{t-h_1+1} отображает на Y_{t-h_1+1} и отображает Y_{t-h_1+1} на X_{t-h_1+1}.

Из того что группа $\overline{A_{t-h_1+1}}$ содержит все изоморфизмы графов $G(X_{t-h_1+1})$ и $H(Y_{t-h_1+1})$, продолжаемые до изоморфизмов графов

$$G\left(\bigcup_{i=t-h_1+1}^{t} X_i\right) \quad \text{и} \quad H\left(\bigcup_{i=t-h_1+1}^{t} Y_i\right),$$

следует, что группа A_E должна содержать все изоморфизмы графов G_{t-h_1} и

H_{t-h_1}, продолжаемые до изоморфизмов графов

$$G\Big(\bigcup_{i=t-h_1}^{t} X_i\Big) \quad \text{и} \quad H\Big(\bigcup_{i=t-h_1}^{t} Y_i\Big),$$

если графы $G\Big(\bigcup\limits_{i=t-h_1}^{t} X_i\Big)$ и $H\Big(\bigcup\limits_{i=t-h_1}^{t} Y_i\Big)$ изоморфны. Если в K_E нет элементов, отображающих X_{t-h_1} на Y_{t-h-1}, то графы $G\Big(\bigcup\limits_{i=t-h_1}^{t} X_i\Big)$ и $H\Big(\bigcup\limits_{i=t-h_1}^{t} Y_i\Big)$ не изоморфны, а значит, графы G и H не будут $[u,v]$-изоморфными и алгоритм заканчивает работу.

Рассмотрим множество $\overline{K_{t-h_1}}$, которое содержит ограничения всех элементов из K_E на множество $X_{t-h_1} \cup Y_{t-h_1}$. Очевидно, $\overline{K_{t-h_1}}$ будет порождать группу $\overline{A_{t-h_1}}$, которая содержит все изоморфизмы графов $G(X_{t-h_1})$ и $H(Y_{t-h_1})$, продолжаемые до изоморфизмов графов $G(\bigcup\limits_{i=t-h_1}^{t} X_i)$ и $H(\bigcup\limits_{i=t-h_1}^{t} Y_i)$.

Поскольку $\overline{A_{t-h_1}}$ является подгруппой группы A'_{t-1}, из теоремы 2.2 следует, что она является 2-группой.

Таким образом, на $(t-1)$-й итерации мы получим множество $\overline{K_1}$. Если множество $\overline{K_1}$ не содержит подстановок, переставляющих вершины из X_1 и Y_1, то графы G и H не будут $[u,v]$-изоморфными. В противном случае они будут $[u,v]$-изоморфными.

Алгоритм 3.1. Вход. Графы G и H из класса $PathSpec_1$. Пусть u – вершина графа G, v – вершина графа H, для которых существуют $PathSpec_1$-разложения $P_u^G = (X_1, \ldots, X_t)$ и $P_v^H = (Y_1, \ldots, Y_{t'})$ графов G и H соответственно.

Выход. Если P_u^G и P_v^H изоморфны, то алгоритм возвращает $true$. В противном случае он возвращает $false$.

Ниже для некоторых операций введены специальные обозначения. Пусть M – некоторое множество. Операция $Get(M)$ возвращает некоторый элемент множества M. Функция $Iso(G,H)$ проверяет, изоморфны ли графы G и H из класса $Spec_1$. Функция $FindAutGraph(G,H)$ находит порождающее множество группы $Aut(G,H)$ для пары изоморфных графов G и H из класса $Spec_1$. Подразумевается, что $FindAutGraph$ использует алгоритм, о существовании которого говорится в предыдущей части. Если K – порождающее множество полиномиальной мощности группы автоморфизмов некоторого графа G, а X и Y – равные по мощности и непересекающиеся подмножества множества вершин G, то операция $IsSwap(K,X,Y)$ определяет, содержится ли в K элемент g такой, что $g(X)=Y$ и $g(Y)=X$. Пусть K – множество подстановок, действующих на множестве $X \times Y$. Функция Restr(K, X) возвращает ограничение K на множество X. Функция Stab(K, Y) строит порождающее множество стабилизатора множества Y в 2-группе, порождаемой множеством K.

1. if $(t \neq t')$ return $false$;
2. if (not $Iso(G(X_t), H(Y_t))$) return $false$;
3. $\overline{K_t} := FindAutGraph(G(X_t), H(Y_t))$;
4. for ($h_1 = 1$ to t) do begin
5.   if (not $Iso(G(X_{t-h_1}), H(Y_{t-h_1})$) return $false$;
6. $K'_{t-h_1} := FindAutGraph(G(X_{t-h_1}), H(Y_{t-h_1}))$;
7. $K := \Big\{ \Big(g, id_{\overline{A_{t-h_1+1}}} \Big) : g \in K'_{t-h_1} \Big\} \bigcup \Big\{ \Big(id_{A'_{t-h_1}}, g \Big) : g \in \overline{K_{t-h_1+1}} \Big\}$;
8. $Edges := \big\{ (u,v) : \{u,v\} \in EG_{t-h_1} \cup EH_{t-h_1}, u \in X_{t-h_1} \cup Y_{t-h_1},$
   $v \in X_{t-h_1+1} \cup Y_{t-h_1+1} \big\}$;
9. $K_{t-h_1} := Stab(K, Edges)$;
10.   if (not $swap(K_{t-h_1}, X_{t-h_1}, Y_{t-h_1})$) return $false$;
11. $\overline{K_{t-h_1}} := Restr(K_{t-h_1}, X_{t-h_1} \cup Y_{t-h_1})$;
12. end;
13. return $true$

Корректность алгоритма 3.1 следует из обсуждений, сделанных перед его формализацией.

Теорема 3.1. *Время работы алгоритма* 3.1 *в худшем случае равно* $O(n^{15})$.

Доказательство. Для каждого h_1, $1 \leq h_1 \leq t$, положим $n_{h_1} = |X_{h_1}| = |Y_{h_1}|$. Оценим временную сложность h_1-й итерации (строки 4–12). Действие в строке 5, как следует из теоремы 2.1, выполняется за время $O(n_{h_1}^{14})$. Строка 6 выполняется за время $O(n^4)$. Поскольку мощности множеств, объединяемых в строке 7, не превышают $O(max(n_{h_1}^4, n_{h_1+1}^4)$, то действие в строке 7 выполняется за $O(max(n_{h_1}^4, n_{h_1+1}^4))$. Действие в строке 9 требует времени $O(n_{h_1}^6 \cdot n_{h_1+1}^6)$. Таким образом, временная сложность каждой итерации

$O(n_{h_1}^1 4 + n_{h_1}^6 \cdot n_{h_1+1}^6 + max(n_{h_1}^4, n_{h_1+1}^4))$. Очевидно, что количество итерации t можно оценить как $O(n)$, где n – количество вершин в каждом из графов.

Суммируя временные сложности всех итераций, получаем, что цикл в строках 4–10 требует времени $O(n^{14})$. Легко видеть, что действия в строках 1–3 не ухудшают эту оценку. Теорема доказана.

Ниже приводится алгоритм проверки изоморфизма графов из класса $PathSpec_1$.

Алгоритм 3.2. Вход. Графы G и H из $PathSpec_1$.

Выход. $true$, если G и H изоморфны, $false$ – в противном случае.

1. Находим вершину u графа G такую, что цепное разложение по расстоянию P_u^G с корнем в u является $PathSpec_1$-разложением графа G (из условия следует, что хотя бы одна такая вершина должна существовать). Перейти к п. 2.
2. $S := VH$; перейти к п. 3.
3. Если $S = \varnothing$, то графы G и H не изоморфны и алгоритм возвращает $false$. В противном случае переходим к п. 4.
4. Берем $v \in S$ и $S := S \setminus v$. Если цепное разложение по расстоянию P_v^H с корнем в v является $PathSpec_1$-разложением цветного графа H, то перейти к п. 5. Иначе перейти к п. 3.
5. С помощью алгоритма 3.1 проверяем, изоморфны ли P_u^G и P_v^H, если да, то графы изоморфны и алгоритм возвращает $true$. Иначе перейти к п. 3.

Теорема 3.2. *Пусть G и H – графы из класса $PathSpec_1$. Алгоритм 3.2 за время $O(n^{16})$ определяет, изоморфны ли графы G и H.*

Эта теорема непосредственно вытекает из теоремы 3.1.

В заключение автор выражает благодарность своему научному руководителю В. А. Баранскому за внимание к работе и замечания, способствовавшие ее улучшению.

1. Асанов М. О., Баранский В. А., Расин В. В. Дискретная математика: графы, матроиды, алгоритмы. Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001.

2. Ахо Х., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.
3. Хопкрофт Дж., Тарьян Р. Изоморфизм планарных графов // Кибернет. сб. Вып. 12. 1975. С. 39–61.
4. Bodlaender H. L. Polynomial algorithms for graph isomorphism and chromatic index on partial k-trees // SWAT 88. [Lecture Notes in Computer Science; Vol. 318]. B.; Heidelberg: Springer, 1988. P. 223–232.
5. Yamazaki K., Bodlaender H. L., de Fluiter B. et al. Isomorphism for graphs of bounded distance width // Algorithms and Complexity. [Lecture Notes in Computer Science; Vol. 1203]. B.; Heidelberg: Springer, 1997. P. 276–287.
6. Babai L., Grigoryev D. Yu., Mount D. M. Isomorphism of graphs with bounded eigenvalue multiplicity // Proc. 14th ACM Symp. on Theory of Computing. San Francisco, 1982. P. 310–324.
7. Каргаполов М. И., Мерзляков Ю. И. Основы теории групп. М.: Наука, 1972.
8. Hoffmann C. M. Group-Theoretics Algorithms and Graph Isomorphism. [Lecture Notes in Computer Science; Vol. 136]. B.; Heidelberg: Springer, 1982.
9. Земляченко В. Н., Корнеенко Н. М., Тышкевич Р. И. Проблема изоморфизма графов // Зап. науч. семинаров ЛОМИ / Мат. ин-т им. В. А. Стеклова, Ленингр. отд-ние. 1982. Т. 118. С. 83–158.
10. Емеличев В. А., Мельников О. И., Сарванов В. И. и др. Лекции по теории графов. М.: Наука, 1990.
11. Цветкович Д., Дуб М., Захс Х. Спектры графов. Теория и применение. Киев: Наук. думка, 1984.

Статья поступила 22.11.2007